# Competitive Air Hockey Database

Alexandria Duell and Rita Berglund
House of Pucks

## Feedback

** Note that the underlined names are links.

### TA Feedback

Good work group 60!
- Kaveenaya Srinivasagam Omkumar

### Peer Review 1: Kevin Phillips

I love the overview, I can tell you put effort into researching your topic! I can easily tell that you are trying to help the Air Hockey league get organized by centralizing player / match statistics.

There are a lot of numbers about how many players are expected at certain tournaments. This adds some fun detail, but trying to put that information together takes readers some effort and they may not get it right. I think it would be helpful to include a rough number of total players and overall matches per year to tie all the information together.

The project describes 5 entities - Players, Locations, Games, Sets, and Matches. Games / Sets / Matches gave me the initial impression that they might be able to be squeezed into one entity, but I think having them separated like this will make it easier to generate more meaningful player statistics.

The Location entity has information about locations where games are played, such as the address, how many tables are available, and the ID of the player who owns it (if applicable). I think it may be beneficial to add a couple extra attributes here - max occupancy and contact info. Both would be helpful in setting up large events. The tables available attribute gives information that is adjacent to venue size, but they are not always related. Contact info would be a quick reference when a venue is chosen, and could be used to store private numbers if a good relationship is built with owners of venues to bypass some or all of the approval process.

The Player entity describes one player in the league, and holds information such as league ID number, name, address, and contact info.

The Matches entity holds most of the information about a match (or set of sets of games), including information about the type of match, the date and location, the players that played, who won, the number of sets to be played, and the status.

The Sets entity holds information about a set of games within a match, including the set number, the match it belongs to, the winner, the start and end times, and the current status.

The Games entity holds information about a game within a set, including the players that played, the points they scored, the winner, the game number, the start and end times, and the current status.

The M:N relationship player_matches should be set up as two separate 1:M relationships, as was done for games. I think that the Locations:Players relationship would be a good candidate for M:N. Many locations would be owned by 0 or 1 players, some home locations may have 2 players or even more players living there, and there could be groups of players that work together to acquire a larger location. The difference here is that Players:Matches always has exactly two 1:M relationships, while Locations:Players can have any number in either direction.

The naming scheme is mostly consistent. The overview contains the entities exactly as recorded later. Both entities and their attributes are all lower case. All entities use plural while *most* attributes use singular. The plural attributes are "num_of_tables" and "notes" in the "locations" entity, "max_sets" and "notes" in the "matches" entity, "player_1_pts" and "player_2_pts" in the "games" entity, and "sets_reffed" in the "match_officials" junction table.

My suggestions would be "available_table_qty", "note", "set_max", "note", "player_1_score", "player_2_score", and "set_reffed_qty" respectively.
All of my feedback is original work

## Peer Review 2: Brandon Connely

Firstly I would like to say that I can really appreciate the premise of this project. As someone who is a participant in some similarly niche activities, I can relate to the frustration of having to juggle multiple third party apps and agree its a really good idea if there can be one centralized location for all data related to an activity to be stored.

That being said, I think you did successfully describe a problem that can be solved with a database-backed website and I think the example you provide is strongly motivated. You also do provide specific facts, such as there being thousands of games annually, the number of states where competitive air hockey is played and some rough player counts for tournaments and other kinds of games. Overall these facts hint at a scenario that well justifies a database but is also at a reasonable scale.

You also do describe over four entities, and each represent a single well defined concept. For instance:

Players- contains useful information about competitors, officials and hosts

Matches- logs relevant information regarding each individual match played

Sets- Contains data about sets which in the context are a set of multiple games within a match

Games- contains the individual games that make up a match

I would have to say I agree with Kevin that Games / Sets / Matches could serve as a single entity, but I don't see any problems with this implementation.

Your entity outlines are excellent and well thought out. You have included the purpose, attributes and data-types and constraints. I would say the constraints are actually one of the strongest parts of this paper. You account for many realism checks that could easily be missed. For example, how the database does not allow a player to play against themselves. Excellent work.

Your ERD looks excellent and logically matches up with what I would expect for this database. Furthermore, you have included at-least one M:M relationship through players and matches.

Finally when it comes to the naming scheme, I found that the thoughts I formulated are identical to those Kevin provided in his review, so I wont waste the time to rewrite the same information. Overall I think this is a excellent draft and will work great as a bases for future work in this class! :)

This review is entirely my own work, no AI tools were used in its formulation.

## Peer Review 3: Aliya Rahman

Hi Group 60,

I love this project and the subculture it is going to power! I'd absolutely say that this overview outlines the database well and paints a solid picture of how the website backed by it would solve a problem. It actually made me Google around to see if I've been living under a rock with respect to the rise of competitive air hockey, because it reads like the first paragraph I'd see on a market research slide for a well-timed product pitch. I'm with you: we need air hockey events to go smoothly, and we need to grow these leagues. Now.

This overview provides clear specifics to explain that we're looking to efficiently store, organize, and serve thousands of game records annually. This is based on well laid out numerical assumptions, e.g. 11 US states are

in scope, with 3 of them showing up as part of an informal "main" tear where scale is larger due to weekly tournaments.

I see seven entities here. Six are master tables: Players (people participating in the network), Locations (places matches are held), Games (games played at a match), Sets (groupings of games), Match Officials (a profile that depends on a Player instance), and Matches (sets of sets of games held at locations with players). One is a junction table: Player Matches, which implements the many to many relationship between players and matches.

The outline is clear on the purpose (summarized above) of each entity, and attribute data types and constraints are appropriate to the nature of the information we need to store. The relationships between them make sense and are implemented correctly on the ERD. The many to many relationship between players and matches captures the need to associate two players to a match without the problems that would arise from simply storing them as "playerA" and "playerB" field on the match object.

Naming conventions are followed around use of single vs plural names and clear naming, and names are consistent across overview, entity documentation, and ERD.

I think the data is well modeled, and really only had one idea for improvement - relevant only if analytics related to people participating is a feature that is wanted. I'm wondering if the "Players" master table might be more appropriately named "People," since its summary says it includes players, officials, and hosts. The thought is mostly for descriptive clarity, but I'm also feeling curious: are any hosts or officials who are *not* players"? Do we want the ability to store them too? Is there a way to know that they someone is an official before they have officiated one of our matches? If it's a no to those questions, then I think the design is complete! And if any of them sound interesting, I wonder if there is usefulness to allowing you to record right on the person's record which roles they are able to, so you could efficiently (i.e. without having to do joins across other tables in your SQL that take time to run) count the number of people who are just players, how many hosts are actually supporting this network, how many officials need trained, etc. Those stats would be cool business intelligence! I was originally going to say that it might be interesting to include an enum or int field to classify their "role" in the database - like 1="Player", 2="Host", 3="Official", but then you wouldn't be able to store more than one role. Instead, perhaps boolean attributes like isPlayer or isOfficial, isHost.

[This review is original content, written without AI assistance.]

**Peer Review 4: Joris Bolsens**

Hi, I think this is a great project. It is very unique and interesting. Unifying the source of truth for a fun and niche sport is great and will probably help it along quite well.

You include many different facts like how many states participate in tournaments, which states contain your primary users, how many users you expect, etc.
I think your proposed DB solution is solid, you have the required amount of entities and they are all distinct items that can be stored as lists. You include all attributes with types, constraints,  and relationships. One thing that might be useful is to add lengths to the varchars, rather than just "varchar" set to "varchar(255)"

I did find a couple things that could be improved. First, you list zip_code as an int, which could be problematic as zip_codes in the US can start with 0, and in the northeast there are zip codes with even two leading zeros. Since you have a 'country' field defined as well, I assume that means you plan to support international locations, in which case Postal Codes often have letters in them as well. Another possible issue is that your players table does not include date of birth. For many statistics, rankings, and bracketing you generally want to be able to group by age range, so having the date of birth for players would be useful. Lastly you mention tournaments in your overview, but this is not reflected in your database outline, I assume a tournament would be a set of multiple matches with many players?

Your relationships are there and make sense you have a many - many relationship with players and matches. I would suggest adding another table for officials, and having a many to many relationship with officials and matches, as I assume an official will officiate more than one match, and you dont currently store much info about the officials at all which could be useful in the future.

This review is original content, written by me alone.

## Actions Based On Feedback

Based on the feedback given, we have chosen to:

- Change attribute names to be all singular for consistent naming conventions.
- Add contact information attributes for the `locations` entity.
- Update the `zip_code` attributes to a varchar(255)
- Change the `players` entity to `people`, in case there are people associated with Air Hockey that are not an Air Hockey player.
- Add DOB information for the `people` entity (originally `players`).
- Deleted `sets_reffed` in the `match_officials` entity.
    - While thinking about changing this attribute to a singular naming convention, we realized that this could be calculated in a SQL query, and does not need to be an attribute in itself.
- Updated varchar attributes to have a character limit.
- Change the relationship between `people` (originally `players`) + `locations` as a location can be owned by more than 1 person, and a person can own more than 1 location.

We are choosing not to merge the `games`, `sets`, + `matches` entities together as individual games are played in some tournaments, while others use the match format. In addition, keeping these entities separate will allow for other forms of game-play easier to implement in the future.

Furthermore, a `tournament` entity is not going to be included in this project as it is not part of the minimal viable product (MVP), and will be added in future iterations. For the MVP, it is assumed that all official game-play will be matches, and no individual sets or games will be played with this database. In future iterations, this may change.

## Overview

Air Hockey is a relatively new competitive sport and is played competitively (or at least has a competitive player living in the state) in 11 states, with the main three being North Carolina, Texas, and Illinois. State tournaments can range from 30-90+ competitors. The world tournaments have on average 90-100+ competitors. There are also weekly tournaments which take place mostly in the main three states and have between 10-40 competitors. Leagues are less common but are formatted as a weekly event that lasts about 2 months per season and usually have between 10-15 competitors.

Many players compete in multiple matches, sets, and games per event. This produces thousands of individual game records annually. Manual spreadsheets are not sufficient as they cannot reliably manage hundreds of players and games per season, thousands of games annually, and historical match and ranking data. Currently, Air Hockey uses multiple third-party apps, which can be confusing for both experts and novices of the sport. It creates unnecessary time upkeep and misinformation on players and tournaments.

A database system is needed to: store player profiles and regional affiliations; record match, set, and game results; support multiple competition formats; enable accurate statistics, rankings, and historical queries; and scale as participation and events grow. One source of truth would help to grow the sport and streamline recording tournaments and matches. It would also allow for novices to become more involved as they will be able to use the app with ease. In the database, we are planning to initially record player information, matches, sets, games, and locations of tables. The database will be designed to be able to expand upon the initial implementation to record tournaments, leagues, and compute statistics.

## Database Outline

### locations

Records the physical venues where air hockey tables are located, matches can be held, and/or tournament locations. This entity tracks both public + private properties.

**Attributes**

`location_id`: int, auto_increment, unique, not NULL, PK
`table_qty`: int, not NULL, default 0
`email:` varchar(255), unique, nullable
`phone_num`: varchar(25), nullable
`street_address_1`: varchar(255), not NULL
`street_address_2`: varchar(255), nullable
`city`: varchar(255), not NULL
`state`: varchar(255), not NULL
`country`: varchar(255), not NULL, default 'USA'
`zip_code`: varchar(255), not NULL
`type_of_address`: enum ('residential', 'commercial', 'club', 'bar', 'other', etc.), not NULL
`location_name`: varchar(255), nullable
`note`: varchar(10000), nullable

**Constraints**

- The combination of `location_name`, `street_address_1`, `street_address_2`, `city`, `state`, `country`, + `zip_code` must be unique

**Relationships**

- M:N relationship with `people`, implemented with a `people_locations` junction table
- 1:M relationship with `matches` , implemented with `location_id` as FK in `matches`

## people

Records all people who participate in or are involved with USAA Air Hockey competitions, including officials and hosts. This is essential for tracking player information, contact details, + statistics.

**Attributes**

`person_id`: int, auto_increment, unique, not NULL, PK
`first_name`: varchar(50), not NULL
`last_name`: varchar(50), not NULL
`gender`: enum ('M', 'F', 'Other', 'Prefer not to say'), nullable
`dob`: date, not NULL
`email:` varchar(255), unique, nullable
`phone_num:` varchar(25), nullable
`street_address_1`: varchar(255), nullable
`street_address_2`: varchar(255), nullable
`city`: varchar(255), nullable
`state`: varchar(255), nullable
`country`: varchar(255), not NULL, default 'USA'
`zip_code`: varchar(255), nullable

**Constraints**

- `email` has a unique constraint when not NULL

**Relationships**

- M:N relationship with `locations`, implemented with a `people_locations` junction table
- M:N relationship with `matches` implemented through `player_matches` junction table
- 1:M relationship with `match_officials` , implemented with `person_id` via `official_person_id` as FK in `match_officials`
- 1:M relationship with `matches` , implemented with `person_id` via `winner_id` as FK in `matches`

- 1:M relationship with `sets` , implemented with `person_id` via `winner_id` as FK in `sets`
- 1:M relationship with `games` , implemented with `person_id` via `winner_id` as FK in `games`

## matches

Records all competitive Air Hockey matches between 2 players. This is the central entity for tracking match information including timing, location, + outcome. A match consists of a max of 3, 5, or 7 sets. The first player to win the required 2 of 3, 3 of 5, or 5 of 7 sets wins the match.

**Attributes**

`match_id`: int, auto_increment, unique, not NULL, PK
`location_id`: int, not NULL, FK references `locations`
`winner_id`: int, nullable, FK references `people`
`set_max`: int, not NULL, default 3, check (set_max in (3, 5, 7))
`faceoff_type`: enum ('standard', 'puck flip'), not NULL, default 'standard'
`start_datetime`: datetime, not NULL
`end_datetime`: datetime, nullable
`match_type`: enum ('challenge', 'tournament', 'league', etc.), not NULL
`match_status`: enum ('scheduled', 'in_progress', 'completed', 'abandoned'), not NULL, default 'scheduled'
`note`: varchar(10000), nullable

**Constraints**

- `end_datetime` must be greater than `start_datetime` when both are not NULL
- `winner_id` must be NULL unless `match_status` = 'completed'
- `set_max` can only be 3, 5, or 7

**Relationships**

- M:1 relationship with `locations` , implemented with `location_id` as FK in `matches`
- 1:M relationship with `sets` , implemented with `match_id` as FK in `sets`
- M:N relationship with `people` , implemented with `player_matches` junction table
- 1:M relationship with `match_officials` , implemented with `match_id` as FK in `` `match_officials` ``
- M:1 relationship with `people`, implemented with `person_id` via `winner_id` as FK in `matches`

## player_matches

Junction table that implements the many-to-many relationship between `matches` + `people`. This entity records all matches per each individual player.

**Attributes**

`player_match_id`: int, auto_increment, unique, not NULL, PK
`player_id`: int, not NULL, FK references `people`
`match_id`: int, not NULL, FK references `matches`
`starting_side`: enum ('left', 'right'), not NULL
`is_winner`: boolean, not NULL, default FALSE

**Constraints**

- Unique constraint on combined `player_id` + `match_id`
- Exactly 2 `player_matches` records must exist per match
- Only one player can have `is_winner` = TRUE per match

**Relationships**

- M:1 relationship with people , implemented with `player_id` as FK in `player_matches`

- M:1 relationship with `matches` , implemented with `match_id` as FK in `player_matches`
- Implements the M:N relationship between `people` and `matches`

## sets

Records individual sets within a match. A set consists of multiple games until a player wins 4 games out of that set. A set contains 4-7 games.

**Attributes**

`set_id`: int auto_increment, unique, not NULL, PK
`match_id`: int, not NULL, FK references `matches`
`winner_id`: int, nullable, FK references `people`
`set_num`: int, not NULL, check (set_num between 1 and 7)
`start_datetime`: datetime, nullable
`end_datetime`: datetime, nullable
`set_status`: enum ('scheduled', 'in_progress', 'completed', 'abandoned'), not NULL, default 'scheduled'

**Constraints**

- Unique constraint on `match_id` + `set_num` - set numbers must be unique within a match
- `end_datetime` must be greater than `start_datetime` when both are not NULL
- `winner_id` must be NULL unless `set_status` = 'completed'
    - A set must contain at least 4 games when completed
- `set_num` cannot exceed the parent match's `set_max` value

**Relationships**

- M:1 relationship with `matches` , implemented with `match_id` as FK in `sets`
- 1:M relationship with `games` , implemented with `set_id` as FK in `games`
- M:1 relationship with `people`, implemented with `person_id` via `winner_id` as FK in `sets`
- 1:M relationship with `match_officials` , implemented with `set_id` as FK in `match_officials`

## games

Records individual games within a set. A game is played until 1 player reaches 7 points. There are at least 4 games in a set, and a max of 7 games.

**Attributes**

`game_id`: int, auto_increment, unique, not NULL, PK
`set_id`: int, not NULL, FK references `sets`
`player_1_id`: int, not NULL, FK references `people`
`player_2_id`: int, not NULL, FK references `people`
`winner_id`: int, nullable, FK references `people`
`player_1_score`: int, not NULL, default 0, check (player_1_score between 0 and 7)
`player_1_score`: int, not NULL, default 0, check (player_1_score between 0 and 7)
`game_num`: int, not NULL, check (game_num between 1 and 7)
`start_datetime`: datetime, nullable
`end_datetime`: datetime, nullable
`game_status`: enum ('scheduled', 'in_progress', 'completed', 'abandoned'), not NULL, default 'scheduled'

**Constraints**

- Unique constraint on `set_id` + `game_num` - game numbers must be unique within a set
- `player_1_id` + `player_2_id` must be different (a player cannot play against themselves)
- `player_1_id` + `player_2_id` must match the 2 players in the parent match

- `Player_1_score` + `player_2_score` must be different if 1 player has 7 points
- `winner_id` must be NULL unless `game_status` = 'completed'
- `winner_id` must be either `player_1_id` or `player_2_id` when not NULL
- `end_datetime` must be greater than `start_datetime` when both are not NULL
- When `game_status` = 'completed', one player must have exactly 7 points

**Relationships**

- M:1 relationship with `sets` , implemented with `set_id` as FK in `games`
- M:1 relationship with `people`, implemented with `person_id` via `player_1_id` as FK in `games`
- M:1 relationship with `people`, implemented with `person_id` via `player_2_id` as FK in `games`
- M:1 relationship with `people`, implemented with `person_id` via `winner_id` as FK in `games`

## match_officials

Records officials (referees + witnesses) for matches and/or sets. Records who officiated, in what capacity, and for what type of entity (`matches` and/or `sets`).

**Attributes**

`match_official_id`: int, auto_increment, unique, not NULL, PK
`official_person_id`: int, not NULL, FK references `people`
`match_id`: int, nullable, FK references `matches`
`set_id`: int, nullable, FK references `sets`
`official_type`: enum ('referee', 'witness'), not NULL

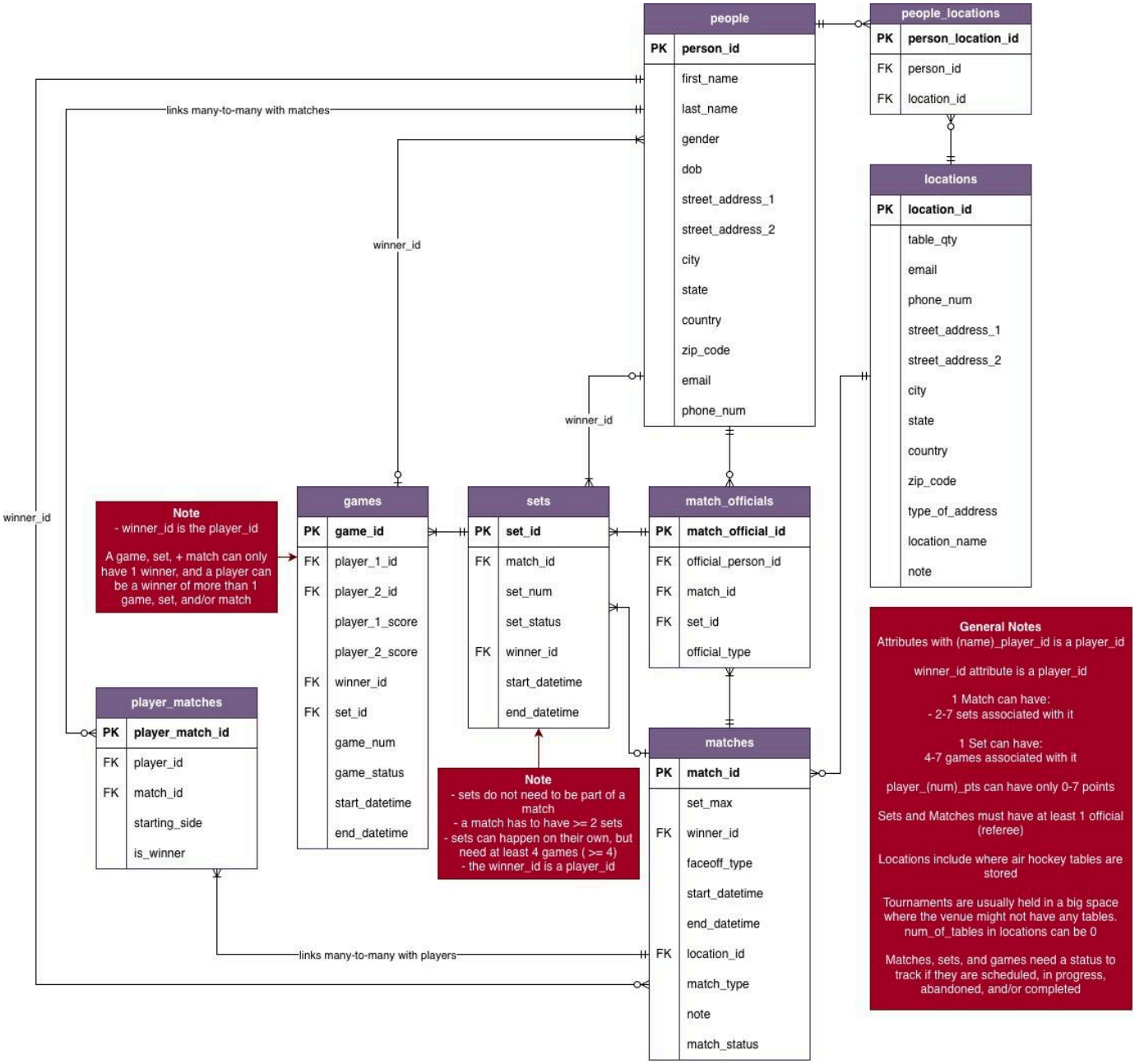
**Constraints**

- If `match_id` is not NULL, then `set_id` must not be NULL - if there is a `set_id`, then `match_id` can be NULL
- `official_person_id` cannot be 1 of the players competing in the match

**Relationships**

- M:1 relationship with people , implemented with `person_id` via `official_person_id` as FK in `match_officials`
- M:1 relationship with `matches` , implemented with `match_id` as FK in `match_officials`
- M:1 relationship with `sets` , implemented with `set_id` as FK in `match_officials`

# Entity-Relationship Diagram




# Citations

All content in this document for the database design is original work from Alex Duell + Rita Berglund. We did research + referenced the USAA Rules + Regulations for challenge matches + games via the NC Air Hockey Players Association website: *FAQs*. NC Air Hockey Players. (2025). https://www.ncairhockeyplayers.com/faqs. Used documents found in the section "Is there a Rulebook for Air Hockey?"